解　説

# 4.2 分散型オペレーティング・システム†

田胡和哉††　益田隆司††

## 1. はじめに

計算機価格の低下が進んでいる．特に，高機能のマイクロプロセッサの出現により，この傾向が著しい．このため，計算機システムの運用形態も質的に変化しつつあり，特に，分散処理化の傾向が強まっている．

分散処理の一つの形態として，分散型オペレーティング・システムによるものがあげられる．分散型オペレーティング・システムは，利用者，および，利用者プログラムが，ネットワーク構造を意識せずにシステムを利用することを可能にする．たとえば，ワークステーションを Ethernet[26] などの LAN で結合したシステムを，分散型オペレーティング・システムを用いて制御する試みが，盛んになされている．分散型オペレーティング・システムを用いることにより，ディスク装置をワークステーション間で共用することができる．また，どのノード上でも均一なジョブ実行環境を提供することが可能であり，ワークステーション間で負荷の分散を図ることができる．

このような機能を持つ分散型オペレーティング・システムを実現するために，オペレーティング・システムの構成方式，性能の改善，耐障害性の向上などに関する研究が行われている．ワークステーションの例に限らず，マイクロプロセッサを利用して機能の向上を図る試みが急速に増加しつつあり，今後，分散型オペレーティング・システムにおいて検討された技術の重要性が増加することが考えられる．本稿では，特に，LAN 用分散型オペレーティング・システムを主体として，その設計の狙い，および，技術について述べる．

## 2. 概　要

### 2.1 分散型オペレーティング・システムとその目的

表-1 に示すように，1980 年代のはじめに，LOCUS[29),34),35),47),48)], Accent*[12),37)], V[9)-11),46)], Eden[7),22)], Amoeba[27),28),43)] システムなどの分散型オペレーティング・システムが相次いで発表された．これを契機として，分散型オペレーティング・システムを用いた計算機システムの研究が盛んになされるようになっている[44)]，これらの多くは，スーパミニコンピュータ，ワークステーションを LAN を用いて結合した分散ハードウェアを対象としている．

ワークステーションの利用法を考えてみる．どのワークステーションからも任意のファイルをアクセスすることができなければ，利用者は，特定のワークステーションしか利用できないことになる．また，すべてのワークステーションにディスク装置を付加することは有効でない．たとえば，ワークステーション用のオペレーティング・システムとして広く用いられている UNIX システム**[39)]は数十メガバイトのディスク容量を必要とし，ワークステーションのディスク装置のかなりの部分を占める．今後も，必要とされるディスクの容量は増加することが予想され，ディスク装置を共同で利用する方が望ましい．

分散型オペレーティング・システムの狙いは，このような，通信，および，資源共有に対する要請を満たすことにある．複数の計算機からなるシステムを制御する機能を持つオペレーティング・システムは，分散型とネットワーク型に大別できる．分散型のシステムは，利用者に分散透明な環境，特に，ジョブの管理機能，および，ファイル・システムに関する分散透明性を提供する機能を持つ．ここで，分散透明とは，利用者がプロセッサの差異を識別できないことを言う．そ

† Distributed Operating Systems by Kazuya TAGO and Takashi MASUDA (Institute of Information Sciences and Electronics, University of Tsukuba).

†† 筑波大学電子・情報工学系

* Accent は，米国カーネギー・メロン大学の登録商標である．

** UNIX は，米国ベル研究所の登録商標である．

表-1 稼働中の代表的分散型オペレーティング・システム

| システム名 | 開発主体 | 稼働初年 | ハードウェア | ネットワーク | OS 構成 | 基本規模 | 特徴 | 参考文献 |
|---|---|---|---|---|---|---|---|---|
| Accent | CMU | 1981 | PERQ WS<br>+ファイル・サーバ | Ethernet | 分散核<br>+UNIX エミュレータ | >100 台 | Spice プロジェクトの一環，大規模な実用システム | 12), 37) |
| Aegis | Applo 社 | ? | 専用 WS | 専用回線 | 特殊 | — | 商用システム，記憶管理をもとにした分散透明性の実現 | 51) |
| Amoeba | Vrije 大学<br>(Netherland) | ? | WS<br>+プロセッサ・プール | Ethernet | 分散核<br>+UNIX エミュレータ | 24 CPU | アトミック・トランザクションを実現したファイル・サーバ<br>オブジェクト指向によるデータ管理<br>プロセッサ・プールの設置 | 27), 28), 43) |
| Cambridge Distributed System | Cambridge 大学 | ? | 個人用計算機<br>+ファイル・サーバ<br>+プロセッサ・プール | Cambridge Ring | 特殊 | ? | プロセッサ・プールの設置 | 31) |
| DEMOS/MP | California 大学 | ? | Z 8000 によるノード/UNIX 上のエミュレータ | 専用回線 | 分散核 | ? | プロセス移動の実現，信頼性重視のファイル・サーバ | 36) |
| Eden | Washington 大学 | 1983 | SUN WS | Ethernet | UNIX 上の分散核 | 16台 | オブジェクト指向によるシステム構成 | 7), 22) |
| LOCUS | California 大学 | 1981 | VAX | Ethernet | 均等構成 | 17台 | アトミック・トランザクションの核内での実現 | 29), 34), 35), 47), 48) |
| V | Stanford 大学 | 1982 | SUN WS | Ethernet | 分散核 | >25台 | 公開市場モデルによるシステム実現<br>プロセス移動による負荷調整機構 | 9)～11), 46) |
| XMS | BNR (Canada) | 1981 | 専用 WS | ? | 分散核 | >100 | 実用システム | 14), 15) |
| Distrubted UNIX | Bell 研 | 1980 | ミニコン | 専用回線 | 特殊 | ? | 仮想回路モデルを実現するインテリジェント・ネットワーク・コントローラ | 24) |

こで，分散型オペレーティング・システムでは，どのプロセッサに付加されたディスク中のファイルも，同一の方法でオープンして入出力できる．また，ファイル名を付ける規則も同一であり，システム全体のファイルが一つのディレクトリ・システムによって管理される．また，利用者プログラムの実行環境も，実行するプロセッサによって変化しない．たとえば，他のプロセッサ上での利用者プロセスの起動，同期なども，同一プロセッサ上で実行するのと同様に行える．そこで，利用者プログラムをどのプロセッサで実行するかも，負荷，および，機能に応じてシステムが決めることができる．

これに対して，ネットワーク型のオペレーティング・システム[38]では，利用者がなんらかの点でプロセッサの配置を意識する必要がある．ネットワーク型のシステムでは，プロセッサを指定してファイル・オープンを行う．また，他のプロセッサ上でのジョブの実行は，遠隔実行コマンドなどの利用者プログラムの補助を必要とする．すなわち，オペレーティング・システムの構造からみると，分散型オペレーティング・システムでは，分散ハードウェア全体を単位として資源管理を行っている．一方，ネットワーク型のシステムでは，オペレーティング・システムは計算機ごとに独立であり，分散処理に関する機能は主として通信機能に限られている．

### 2.2 分散型オペレーティング・システムの発展経緯

小型機からなる分散型オペレーティング・システムを系統的に実現する方式は，Brinch Hansen の試み[5]にその原形をみることができる．そこでは，システムを通信で結合されたプロセスの集合によって設計する方式が提案され，このような設計法が分散型オペレーティング・システムの実現に適応しやすいことが指摘されている．実際に利用可能な分散処理システムの例として，CMU の C. mmp システム，および，Hydra オペレーティング・システム[50]があげられる．その後，Medusa[33]，Muss[13]，Roscoe[41]，Star OS[18]，Resource Sharing UNIX，RIG[3),4)]などのシステムが開発されてきた．また，我が国でも，1970年代に ACE[17]，KOCOS[19]，TECNET[45]システムなどをはじめとして，多くのシステムが開発されてきた[1),25),30)]．

これらのシステムの狙いとして，小型機を用いて大型機なみの性能を得ることがあげられる．これらの試みにより，分散型オペレーティング・システムの実現方式に関する基礎的な技術が確立された．

## 3. システム構成と実現技術

### 3.1 システム構成とその狙い

現在試みられているシステム構成は，大きく分けて，計算機を対等に結合するものと，専用機能を持つサーバを用いるものがあげられる．Popek らは，参考文献34)において，前者を統合モデル (integrated model)，後者をサーバ・モデル (server model) と名付けている．ここでも，この名称を用いる．

スーパミニコンピュータ，ディスク装置付きのワークステーションなどの，単独で動作可能な計算機を対等に結合し，分散型オペレーティング・システムを用いて制御することにより，同一地点からどの計算機も均等に利用することができるようになる．これにより，ネットワークの使い勝手の向上，および，計算機間での負荷の調整を図ることができる．

サーバ・モデルによるシステム構成の例を図-1 に示す．ディスク装置のない（ディスクレス）ワークステーション，および，ファイル・サーバ，プリンタ・サーバなどのサーバからなる．この構成の目的は，計算機資源を共有することにより，実現効率の改善を図ることにある．特に，ファイル・サーバに関して多くの研究がなされており，個々のワークステーションにディスク装置を付加するのに比べてより良好な価格／性能比が得られることが実測によって確認されている[23]．

サーバ・モデルによる構成において，ジョブ・サーバを設けることにより，絶対性能の向上を図る試みも始まっている[43]．たとえば，高性能の専用計算機，あるいは，プロセッサのプールを設けて，コンパイルな

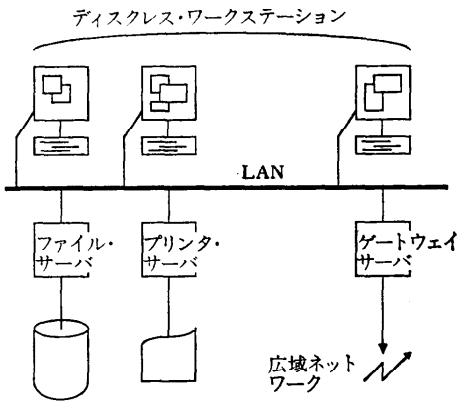



図-1 サーバ・モデルによる分散処理システムの例

どのバッチ的なジョブを集中的に処理する方式が試みられている．しかしながら，現時点では，このような構成がどの程度有効であるかは明らかにされていない．

ネットワークの結合に関しても，種々の構成が試みられている．特に，サーバ・モデルによる構成では，単一のネットワークの規模には制限があり，上位のネットワークが利用される．これを用いて分散透明性を保ったままシステムの規模を拡張する方式[44]と，開放システム間結合として利用する方法[10]がある．

### 3.2 実現技術

#### 3.2.1 構成技術

分散型オペレーティング・システムの実現方式として，主に，たとえば UNIX システムなどの既存のオペレーティング・システムを分散型に改造する方法と，分散核 (Distributed Kernel) による方法があげられる．

計算機間を対等に結合する統合モデルによる方法では，各計算機には同一構成のオペレーティング・システムが配置されるので，システムの実現効率が特に高い必要はない．そこで，既存のオペレーティング・システムを必要に応じて改造する方法がとられる．

サーバ・モデルによる方法では，計算機ごとに異なる機能を配置する必要が生ずるので，既存の設計方式は適用しにくい．そこで，分散核による方法を用いる例が多い．分散核は，プロセス間通信，記憶管理，物理入出力の管理などを実現しており，個々の計算機に同一のものが配置される．既存のオペレーティング・システムにおける，ファイル管理，プログラムの実行管理などは，核外の，ソフトウェア・サーバによって実現されている．ソフトウェア・サーバは，単一の利用者プロセスまたはその集合によって実現されている．一般に，階層的な依存関係を持つサーバによるシステム構成を，サーバ・クライエント・モデル (server client model) とよぶ．分散核が分散透明なプロセス間通信の実行環境を実現することにより，ソフトウェア・サーバは，任意の計算機からの要求を処理することができる．プロセス間通信方式として，RPC (Remote Procedure Call) が広く用いられている．

図-2 に，代表的な分散核の一つである V kernel 上における，プログラム起動過程を示す[46]．新たなプログラムを実行しようとするプロセスは，プログラム管理プロセスにその生成を依頼する(①)．このプロセスは，V kernel 内にある特殊プロセスに記憶領域，および，プロセスの登録を依頼し(②)，ファイル・サーバ・ノードにあるファイル管理プロセスにプログラムのロードを依頼する(③)．起動を依頼したプロセスは，ロードされたプログラムに引数をセットした後，生成されたプロセスに初期化メッセージを送る(④)ことにより，プログラムの実行を開始する．プログラム管理，ファイル管理，および，核内の特殊プロセスが，システムによって用意されたソフトウェア・サーバである．このようなソフトウェア・サーバを計算機間に分散することにより，サーバ・モデルによるシステムの制御を行うことができる．

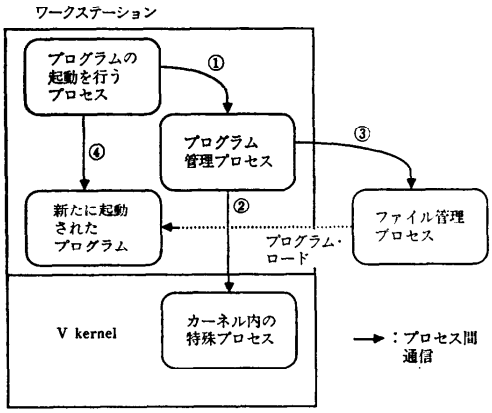



図-2 Vシステムにおけるプログラム実行

RPC は，同期型の通信方式である[2),6)]．分散核は，通信のためのプリミティブを実現しており，システム・コールと同様の方法でプロセスがそれを利用する．RPC は，SEND, RECEIVE, REPLY プリミティブによって実現される．SEND を，通信相手のプロセスの識別子，および，通信相手に渡すべきデータを引数として呼び出すことにより通信呼出しが開始される．SEND を呼び出したプロセスは，通信が終了するまで停止する．RECEIVE により通信の受けつけが行われ，通信データが，受けつけたプロセスに渡される．このプロセスが REPLY を発行することにより通信が終了し，SEND を発行したプロセスが実行可能になる．RPC では，通信データを両プロセスの論理アドレス空間の間で直接転送することが可能である．このため，分散核がバッファリングを行う必要がなくなり，非同期型の通信に比べて実現が容易でオーバヘッドも少なくすることができる．

#### 3.2.2 データ管理技術

分散型オペレーティング・システムにおけるデータ管理上の問題点として，信頼性，および，アクセスの

並行性の制御があげられる．分散処理システム全体としてみると，その一部分が故障する確率は，単一プロセッサ・システムが故障する確率に比べて高い．一方，システム全体が故障して全く処理が行えなくなる確率は，単一プロセッサ・システムの方が高い．そこで，分散処理システムでは，部分的な故障による影響を少なくすることが必要であり，また，その有効性が高い．部分的な故障による影響を少なくすること，および，性能を向上することを目的として，同一のファイルのコピーを複数の計算機上に置くことがしばしば行われる．これらのコピーを矛盾なく変更するために，アクセスの並行性の制御が必要である．

これらの要請は，データベース管理システムにおける要請と類似している．そこで，データベース管理用に開発された技術の導入が図られている．代表的なものとして，アトミック・トランザクション機構があげられる[20),21),28),29),32),48)]．アトミック・トランザクション機構では，図-3 に示すように，一連のファイル入出力をまとめてトランザクションとして扱う．トランザクションの開始および終了は，開始およびコミット命令によって利用者が宣言する．トランザクション中にはファイル本体の変更は行われず，変更部分を別個に保持する．トランザクションの終了後検証が行われ，変更が矛盾なく行われていれば，変更前のファイルを変更の加えられたものと一時に交換する．変更中のシステム・ダウンなどにより，検証に矛盾が生じた場合には変更部分はすべて廃棄される．また，利用者がコミット命令の代わりにアボート命令を発行した場合にも廃棄される．これにより，システム，および，利用者プログラムの処理が正しく行われた場合に限り，ファイルの変更が行われる．

```
fid=open(file_name);        ファイルのオープン
.....
tbegin(fid);                トランザクションの開始
.....
write(fid, buffer);         ファイルへの書込み
.....                       コミット命令
write(fid, buffer);
.....
commit(fid);
トランザクション
```

図-3　アトミック・トランザクション

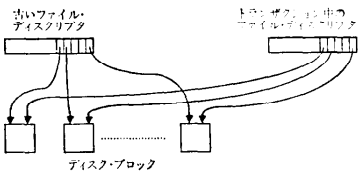
アトミック・トランザクション機構により，各ファイルの内容は，トランザクションの終了ごとに非連続的に更新される．そこで，ファイルのバージョン管理が容易になる．また，トランザクション間の相互排除，あるいは，アクセスが衝突した場合にはトランザクションを再度行う[20),28)]ことにより，アクセスの並行性を制御することができる．さらに，複数コピーの論理的同時変更を実現することができる．

アトミック・トランザクション機構は，図-4(a)に示すように，トランザクションの開始時に新たなファイル・ディスクリプタを複製することにより実現できる．トランザクション中は，このディスクリプタを用いてファイルにアクセスする．図-4(b)に示すように，変更のあったディスク・ブロックのみを割り付け，他のブロックは共有する．古いディスクリプタを用いることにより，トランザクション中にも他の利用者プログラムがファイル入力を行うことができる．また，トランザクション終了後もこのディスクリプタを保存することにより，古いバージョンが保持できる．

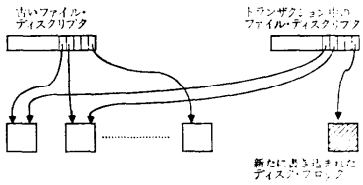
### 3.2.3 性　能

分散型オペレーティング・システムを用いることにより，並列処理，および，機能分散による性能の向上を期待することができる．一方，通信によるオーバヘッドが増加することが予想される．現時点では，汎用の分散型システムにおいて並列処理による性能向上が図られた例はまれである[33)]．機能分散による性能の向上についても，具体的な報告はなされていない．一方，分散化にともなうオーバヘッドの削減については種々検討されている．

分散核による方式では，プロセス間通信のオーバヘ




(a)　ディスクリプタの複製




(b)　ディスク・ブロックの交換

図-4　アトミック・トランザクションの実現

ッドが重大である．その改善が試みられている．たとえば，Vシステムの通信プリミティブでは，SEND 通信プリミティブによって転送されるのは，32 byte の固定長パケットである．大量のデータを転送する場合には，パケットを用いてポインタのみを転送し，呼出しプロセスが停止している期間内に別個のプリミティブを用いて転送を行う方法をとっている．Accent システムでは，仮想記憶機構とプロセス間通信機構を組み合わせて実現する方法をとっている[12]．

いま一つの問題点として，計算機間での負荷のバランスがあげられる．特に，ワークステーションのネットワークでは，性能の高くない計算機が多数存在することになり，負荷のバランスが重要である．このため，実行中のプロセスを計算機間で移動する (process migration) ための技術の開発が行われている．たとえば，Vシステムでは，移動期間中計算機の状態を凍結する方法について考察している[46]．

## 4. 研究の現状と今後の課題

### 4.1 現 状

表-1 に示した分散型システムの多くは，複数のサイトで実用に供されており，同時に改良が進められている．これらのほかにも，ニューカッスル大学の Newcastle Connection[8]，MIT の Argus[32]，ニューヨーク州立大学の MICROS[49] システムなどが発表されている．また，Xerox 社の Palo-Alto 研究所の研究[6]が知られている．筆者らも Agora-proto システム[42]を開発した．主要なものについて，システムの狙い，用いている技術について述べる．

(1) Accent システム[12),37)]

カーネギー・メロン大学において，統一的な計算機利用環境を提供することを目的として行われた，Spice プロジェクトの一環として開発された．主に，PERQ ワークステーションを対象とした分散核であり，大規模なシステムを構成している．この上で，Sesami ファイル・サーバなどのシステム機能，および，種々のアプリケーション・ソフトウェアが開発された．

(2) Amoeba システム[27),28),43)]

オランダの Vrije 大学において開発されている分散型オペレーティング・システムである．ジョブを集中的に処理するプロセッサ・プールを備えている．また，記憶媒体に関して階層構造を持つファイル・サーバを実現している．さらに，ワークステーションによる広域ネットワークへの適用を検討している．このような環境では，システム管理ソフトウェアは各計算機のハードウェア，核を信頼することが不可能であり，それらに依存しない方法で資源へのアクセス権を管理する必要がある．そこで，オブジェクトを指向した方法により統一的にデータを管理し，暗号化によってアクセス権を保護している．

(3) Eden[7),22)]

Eden システムは，ワシントン大学で開発されている．オブジェクトを指向した環境をハードウェアによって実現した，インテル社の iAPX 432 プロセッサ上で動作するシステムの実現を目標として出発した．このため，オブジェクトを指向した方式により設計されている．各オブジェクトには複数のサーバが付加されており，それらは常にアクティブである．システムは，オブジェクトの集合として実現されており，利用者が直接アクセスすることができる．オブジェクトは，プロセッサ間を移動することができる．現在では，UNIX オペレーティング・システムの上に構築されている．

(4) LOCUS システム[29),34),35),47),48)]

LOCUS システムは，カルフォルニア大学で開発されており，DEC 社の PDP 11 コンピュータのネットワークを出発点としている．現在では，スーパミニコンピュータに移行されている．LOCUS システムは，単独で動作可能な計算機を対等に結合することを目的としており，UNIX システムと上位互換性を持つ．

システムの目的にあわせて，安全で効率のよいファイル・システムの実現に主眼がおかれている．また，実行時にネットワーク構造を変更することのできる機能を備えており，一部の計算機がダウンしても運用を継続し，修理後にその計算機をネットワークに復帰させることができる．ファイル・システムは，アトミック・トランザクション機能を備えており，ファイルの入出力に加えて，トランザクションの開始，コミット用のシステムコールを実現している．これをもとに，ファイルの一部分に対する同時アクセスのロック，トランザクションの入れ子を実現している．また，単一ファイルの複数コピーに対する同時更新機能を持つ．

オペレーティング・システムは，既存の UNIX システムの構造をそのまま継承している．性能を重視し，機能ごとに，問題指向 (problem oriented) のプロトコルを設定し，各階層が通信を直接実行する．

（5） Vシステム[9)~11),46)]

Vシステムは，スタンフォード大学において開発された，分散核による分散型オペレーティング・システムである．Vシステムは，ディスクレスの SUN ワークステーションとファイル・サーバを主体とするネットワークから構成され，各計算機上には V kernel が配置されている．Vシステムの狙いは，公開市場モデル (open market model) によってシステムを構成することにある．分散核は，計算機ハードウェアにおいてボード間を結合しているマザーボードにおける機能を，ソフトウェアに対して実現している．分散核はプロセスを実現するために必要な機能のみを実現しており，サーバは必要に応じて追加できる．

ファイル・サーバを用いた遠隔ファイルアクセスの性能に関して詳細な報告がなされている[9),23)]．

### 4.2 今後の課題

（1） 技術的課題

LAN 上で，分散型オペレーティング・システムを実現する上での，今後の課題として，性能，信頼性，および，実現効率の改善があげられる．具体的には，以下の課題があげられる．

1） プログラム実行環境の信頼性の改善．システムの故障によってプログラムの実行が影響を受けるのを防ぐための技術が十分でない．チェックポイントを設けてダンプをとる方法は，高価である．

2） プロセッサ間での負荷の調整．

3） ジョブ・サーバの有効性の評価．

4） プロセス間通信に関する必要性能の解析，および，改善方式の開発．プロセス間通信の性能は，たとえば，ハードウェア上の補助により，さらに改善できる余地がある．どのような方法を用いて，どの程度改善すべきかを明らかにする必要がある．

5） 実現環境の整備．多数のプロセッサからなるシステムの開発は，単一プロセッサ用のシステムの開発に比べてはるかに難しい．デバッグ方式を検討する必要がある．

6） 設計方式の改善．通信で結合されたプロセスによる並行動作システムの論理構造の記述方式，デッドロックの回避アルゴリズムの開発などが必要である．

（2） 構成上の課題

1970年代から続く努力により，小型，超小型機向きの分散型オペレーティング・システム実現に関する，技術的めどが立ったといってよい．しかしながら，既存の，たとえば TSS の技術に比較して，適用範囲も狭く，システムの形態も一定していない．試験的なシステムを利用することにより，分散型オペレーティング・システムを用いたシステムの運用，利用方法に関する知見の蓄積が行われつつある段階であると考えられる．

たとえば，カーネギー・メロン大学において，Spice に続く Andrew プロジェクト[16)]では，Accent システムは採用されなかった．かわりに，共用のファイル・サーバ[40)]が開発され，各ワークステーションの制御には UNIX オペレーティング・システムが用いられている．主な理由として，システムの規模があげられる．Andrew プロジェクトでは，最終的には 5,000 台程度のワークステーションを結合することを想定しており，安全性，信頼性の点から，分散型のシステムでは制御することが困難である．

この例から考えられるように，今後の一つの方向として，小規模な分散型システムをネットワーク型のオペレーティング・システムを用いて結合するなどの，階層構造の検討があげられる．そこでは，分散型システムは，ワークステーションの持つ使い勝手の良さと，TSS システムの持つ良好な効率を両立させるために用いられる．

## 5. むすび

分散型オペレーティング・システムの目的，構成技術，実現例などについて述べた．

個々の企業，研究所，大学などで利用される計算機の台数が急速に増加していることから，LAN における分散透明性の実現が現実のニーズとして認識されつつある．また，マイクロプロセッサの出現により，分散型オペレーティング・システムによる高機能化が商業的にも成功する可能性が生じてきた．現時点での，最も大きな課題は，どのような構成で，どのような技術を用いてこのニーズを満たすべきかを明らかにすることにある．これに基づいて，広く利用できる分散型オペレーティング・システムが近く出現する可能性も高い．その時点で，分散型オペレーティング・システムに関する各技術の位置付けが定まるものと予想される．

## 参考文献

1) 相磯秀夫：機能分散型計算機システム，情報処理，Vol. 18, No. 4, pp. 325-333 (1977).
2) Andrews, G. R. and Scheider, F. B.: Concepts

and Notations for Concurrent Programming, Comp. survey, Vol. 15, No. 1, pp. 3-43 (1983).
3) Ball, J. E. et al.: RIG Rochester's Intelligent Gateway: System Overview, IEEE Transactions on Software Engineering, Vol. 2, No. 4, pp. 321-328 (1976).
4) Ball, J. E. et al.: Perspectives on Message-based Distributed Computing, Proceedings of 1979 Networking Symposium, IEEE, pp. 46-51 (1979).
5) Brinch-Hansen, P.: The Nucleus of a Multiprogrammed System, CACM, Vol. 13, No. 4, pp. 238-250 (1970).
6) Birrel and Nelson, B. J.: Implementing Remote Procedure Calls, Transaction on Computer Systems, Vol. 2, No. 1, pp. 39-59 (1983).
7) Black, A. P.: Supporting Distributed Applications: Experience with Eden, Proceedings of the Tenth Symposium on Operating Systems Principles, pp. 181-193 (1985).
8) Brownbridge, D. R., Marshall, L. F. and Randell, B.: The Newcastle Connection or UNIX of the World Unite!, Software Practice and Experience, Vol. 12, pp. 1147-1162 (1982).
9) Cheriton, D. R. and Zwaenepoel, W.: The Distributed V kernel and its Performance for Diskless Workstations, Proceedings of the Ninth Symposium on Operating Systems Principles, pp. 129-140 (1983).
10) Cheriton, D. R.: The V kernel: A Software Base for Distributed Systems, IEEE Software, Vol. 1, No. 2, pp. 19-43 (1984).
11) Cheriton, D. R. and Mann. T. P.: Uniform Access to Distributed Name Interpretation in the V System, Proceedings of the Fourth International Conference on Distributed Computing Systems, IEEE, pp. 290-297 (1984).
12) Fitzgerld, R. and Rashid, R.: The Intergration of Virtual Memory Management and Inerprocess Communication in Accent, ACM Transactions on Computer Systems, Vol. 4, No. 2, pp. 13-14 (1985).
13) Frank, G. K. and Theaker, C. J.: The Design of the MUSS Operating System, Soft. Pract. Exper., Vol. 9, pp. 599-620 (1979).
14) Fridrich, M. and Older, W.: The Felix File Server, Proceedings of the Eighth Symposium on Operating Systems Principles, pp. 37-46 (1981).
15) Gammage, N. and Casey, L.: XMS: A Rendezvous-Based Distributed System Software Architecture, IEEE Software, Vol. 2, No. 3, pp. 9-19 (1985).
16) Horris, J. H. et al.: Andrew: A Distributed Personal Computing Environment, CACM, Vol. 29, No. 3, pp. 184-201 (1986).
17) 飯塚 肇他：モジュール型複合計算機 (ACE) の試み，情報処理，Vol. 20, No. 4, pp. 314-322 (1979).
18) Jones, A. K. et al.: Star OS, a Multiprocessor Operating System for the Support of Task Forces, Proceedings of the Seventh Symposium on Operating Systems Principles, pp. 117-127 (1979).
19) 上林憲行，相磯秀夫：プロセス間通信機能指向ミニコンピュータ複合体 KOCOS のアーキテクチャ，情報処理，Vol. 20, No. 4, pp. 307-313 (1979).
20) Kung, H. T. and Robinson, J. T.: On Optimistic Methods for Concurrency Control, Vol. 6, No. 2, pp. 213-226 (1981).
21) Lampson, R. W.: Atomic Transactions, Distributed Systems-Architecture and Implementation, Springer-Verlag, Berlin and New York, pp. 246-265 (1981).
22) Lazowska, E. et al.: The Architecture of the Eden System, Proceedings of the Eighth Symposium on Operating Systems Principles, pp. 148-159 (1981).
23) Lazowska, E. et al.: File Access Performance of Dikless Workstations, ACM Transactions on Computer Systems, Vol. 4, No. 3, pp. 238-268 (1986).
24) Luderer, G. W. R. et al.: A Distributed UNIX System Based on a Virtual Circuit Switch, Proceedings of the Eighth Symposium on Operating Systems Principles, pp. 160-168 (1981).
25) Maekawa, M. et al.: Experimental Polyprocessor System (EPOS)-Architechture, Proceedings of Sixth Annual IEEE Symposium on Computer Architechture, pp. 188-195 (1979).
26) Metcalfe, R. M. and Boggs, D. R.: ETHERNET: Distributed Packet Switching for Local Computer Networks, CACM, Vol. 9, No. 7, pp, 395-404 (1976).
27) Mullender, S. J. and Tanenbaum, A. S.: Protection and Resource Control in Distributed Operating Systems, Computer Networks, Vol. 8, pp. 421-432 (1984).
28) Mullender, S. J.: A Distributed File Service Based on Optimistic Concurrency Control, Proceedings of Tenth ACM Symposium on Operating Systems Principles, pp. 51-62 (1985).
29) Mueller, E. T., Moore, J. D. and Popek, G. P.: A Nested Transaction Mechanism for LOCUS, Proceedings of Ninth Symposium on Operating Systems Principles, pp. 71-89 (1983).
30) Murakami, K., Nishikawa, S. and Sato, S.: Poly-processor System Analysis and Design, Proceedings of Fourth Annual Symposium on

Computer Architecture (1977).
31) Needham, R.M. and Herbert, A.J.: The Cambridge Distributed Computing System, Addison-Wesely, Massachusetts (1982).
32) Oki, B.M., Liskv, B.H. and Scheifler, R.W.: Reliable Object Storage to Support Atomic Actions, Proceedings of the Tenth Symposium on Operating System Principles, pp. 147-159 (1985).
33) Ovsterhout, J.K., Scelza, D.A. and Sindhu, P.S.: Medusa: An Experience in Distributed Operating System Structure, CACM, Vol. 23, No. 2, pp. 92-105 (1980).
34) Popek, G. et al.: LOCUS: A Network Transparent, High Reliability Distrubuted System, Proceedings of the Eighth Symposium on Operating Systems Principles, pp. 169-177 (1981).
35) Popek, G.J.: The LOCUS Distributed System Architecture, The MIT Press, Cambridge USA (1986).
36) Powell, M.L. and Miller, B.P.: Process Migration in DEMOS/MP, Proceedings of the Ninth ACM Symposium on Operating Systems Principles, pp. 110-119 (1983).
37) Rashid, R.F. and Robertson, G.G.: Accent: A Communication Oriented Network Operating System Kernel, Proceedings of the Eighth Symposium on Operating Systems Principles, pp. 64-75 (1981).
38) Redell, D.D. et al.: Pilot: An Operating System for a Personal Computer, CACM, Vol. 23, No. 2, pp. 81-92 (1980).
39) Ritchie, D.M. and Thompson, K.: The UNIX Time-Sharing System, CACM, Vol. 17, No. 7, pp. 365-375 (1974).
40) Satyanarayanan, M. et al.: The ITC Distributed File System: Principles and Design Proceedings of the Tenth Symposium on Operating Systems Principles, pp. 35-50 (1985).
41) Solomon, M.H. and Finkel, R.A.: The Roscoe Distributed Operating System, Proceedings of the Seventh Symposium on Operating Systems Principles, pp. 108-114 (1979).
42) 高野，田胡，福田，益田：分散型オペレーティング・システムの構成法に関する一提案，第33回オペレーティング・システム研究会資料 (1986).
43) Tanenbaum, A.S. and Mullender, S.J.: An Overview of the Amoeba Distributed Operating System, Operating System Review, Vol. 15, pp. 51-64 (1981).
44) Tanenbaum, A.S. and Renesse, R.V.: Distributed Operating Systems, Acm Comp. survey, Vol. 17, No. 4, pp. 419-470 (1986).
45) Tanaka, H. and Moto-Oka, T.: Network Oriented Operating System-Process Control and Distributed File in TECNET, Pacnet Symp., 171 (1975).
46) Theimer, M.M., Lantz, K.A. and Cheriton, D.R.: Preemptable Remote Execution Facilities for the V-System, Proceedings of the Tenth Symposium on Operating Systems Principles, pp. 2-12 (1985).
47) Walker, B. et al.: The LOCUS Distributed Operating System, Proceedings of the Ninth Symposium on Operating Systems Principles, pp. 49-70 (1983).
48) Weinstein, M.J. et al.: Transactions and Synchronization in a Distributed Operating System, Proceedings of the Tenth Symposium on Operating Systems Principles, pp. 115-126 (1985).
49) Witte, L.D. and VAN Tilborg, A.M.: MICROS, a Distributed Operating System for MICRONET, a Reconfigurable Network Computer, IEEE Trans. Comput. C-29, pp. 1133-1144 (1980).
50) Wulf, W. et al.: HYDRA: The Kernel of a Multiprocessor Operating System, CACM, Vol. 17, No. 6, pp. 37-345 (1974).
51) 山村紀夫：アポロ DOMAIN，情報処理，Vol. 25, No. 2, pp. 138-143 (1984).

(昭和62年1月24日受付)